multi-SAT: An Adaptive SAT Solver

Sajjad Siddiqi
Jubail University College
Jubail Industrial City, Saudi Arabia
siddiqis@ucj.edu.sa

Jinbo Huang
Australian National University
Canberra, Australia
jinbo.huang@anu.edu.au

Abstract

Mainstream clause learning SAT solvers use decision heuristics that are based on incrementing the scores of variables involved in the learning of conflict clauses. Such a decision strategy emphasizes locality, and accounts for much of the success of modern SAT solvers. However, in some cases the resulting bias toward limited portions of the search space can be detrimental to efficient solutions. The present work has originated on the assumption that efficient solutions can be extended to a greater class of problems if the decision strategy is made more adaptive. In other words, bias in different directions may be beneficial for different types of problems. In this spirit, we propose a new decision framework for SAT that incorporates multiple decision heuristics (for both the initial ordering of variables and their dynamic reordering), periodic assessment of their effectiveness during search, and mechanism to switch on the fly between them based upon the outcomes of their assessments. Incorporating this framework we build a solver named MULTI-SAT based upon GLUCOSE (Audemard and Simon 2009) and submit to SAT-Race-2015 for evaluation.

Detailed Description

It is well known that a decision heuristic may perform well on only a class of problems and no heuristic is expected to work well on all problems. The motivation behind this work is to somehow combine the strengths of different heuristics into a single solver to solve more problems. Since each heuristic is expected to perform well on a particular set of problems, maintaining a reasonably good set of heuristics may enable efficient solutions of a larger set than an ordinary SAT solver can achieve. The same idea has previously motivated researchers to develop portfolio-based SAT solvers (Leyton-Brown et al. 2003a; 2003b; Liberto et al. 2013), which maintain portfolio of several SAT solvers and select the best solver to run on an input problem using a heuristic that is driven by features of the input problem, where such a heuristic is empirically constructed. The work presented in this paper is similar in several ways to porfolio approach but is fundamentally different in terms of selection heuristic.

The new adaptive framework, builds upon the ideas of Shacham and Yorav (Shacham and Yorav 2006), which can maintain a set of decision heuristics each with particular schemes for initial and dynamic variable ordering. The framework allows switching amongst the decision heuristics on the fly based upon an estimate of how good the heuristic is expected to perform on a particular SAT instance at a particular time. For this purpose it periodically performs sample executions of each individual heuristic on the given instance and estimates which heuristic is likely to be more effective at a particular time in the varying search conditions. Several criteria can be used to measure the effectiveness of a particular heuristic. We investigate criteria based upon known ideas of *satisfaction power* (the tendency of a heuristic toward satisfying the clauses in current clause database), *proof width* (Ben-Sasson and Wigderson 2001), *LBD measure* (Audemard and Simon 2009), and a measure of solver progress (Audemard and Simon 2012) (used in the restart strategy of GLUCOSE).

Given a set of heuristics, the adaptive mechanism works as follows. Since it is impossible to know in advance which heuristic would perform best on a particular SAT instance. Therefore sample executions of all heuristics are performed periodically and predictions about their effectiveness are made. For this purpose, at the beginning and at certain normal restarts the solver goes into an *assessment mode* in which a sample execution of every decision heuristic is performed. During assessment mode a decision heuristic is allowed to run for only c number of conflicts and then a restart is forced. At the restart the control is switched to the next heuristic. The assessment mode is finished once all heuristics have been executed, at which place the best heuristic is selected (according to criteria mentioned above) and executed for a number of conflicts determined by the heuristic's own restart policy.

Our implementation mechanism allows that the scores, phases, and statistics maintained by decision heuristics are isolated from each other. The execution of a particular heuristic does not directly change the scores, phases, and statistics maintained by other heuristics. Also, when a heuristic is executed another time during search it uses its old scores that were saved during its last execution. However, the clause database and the inference engine of the solver are shared by all heuristics. As a result, when a heuristic decides to perform clause database reduction it may delete clauses learnt by other heuristics.

We have incorporated 10 arbitrary decision heuristics. A decision heuristic for satisfiability is characterised by particular schemes for initial and dynamic variable ordering, phase saving / selection, score decaying, restarts, clause learning, and clause database reduction. Variations in these schemes lead to different decision heuristics. However, in the current implementation all heuristics, although somewhat independent from each other, work in the same way in all aspects as the default heuristic in GLUCOSE with just two modifications.

1. Every heuristic differs from others in only the initial variable ordering.
2. Every heuristic, in addition to its normal schedule for restarts, forces a solver restart after every time clause database is reduced. At this forced restart new samples of the decision heuristics are also taken in view of the potentially significant change in the search conditions due to the clause database reduction (modern SAT solvers often employ an aggressive clause deletion policy). These samples are taken in addition to the normal sampling strategy of the solver (described above).

Initial variable orders are generated using simple topology-based methods by performing depth-first traversals of the CNF graph (Rice and Kulhari 2008). Such schemes tend to keep the variables that are topologically close to each other together in the variable order. By using different initial variable orders in different heuristics, we hope that even only the differences in the initial variable orders will lead each heuristic to different portions of the search space and will help break the bias generated by a single initial variable order in a traditional solver.

In SAT-Race-2015 we submit two versions of MULTI-SAT namely MULTI-SAT-G2_0 based on GLUCOSE 2.0, and MULTI-SAT-G2_2 based on GLUCOSE 2.2. MULTI-SAT-G2_0 employs a heuristic selection based on satisfaction power as well as a measure of solver progress, while MULTI-SAT-G2_2 employs a heuristic selection based on LBD measure and proof width.

References

Audemard, G., and Simon, L. 2009. Predicting learnt clauses quality in modern sat solvers. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 399–404. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Audemard, G., and Simon, L. 2012. Refining restarts strategies for SAT and UNSAT. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP)*, 118–126. Berlin, Heidelberg: Springer-Verlag.

Ben-Sasson, E., and Wigderson, A. 2001. Short proofs are narrow–resolution made simple. *Journal of the ACM* 48:149–169.

Leyton-Brown, K.; Nudelman, E.; Andrew, G.; McFadden, J.; and Shoham, Y. 2003a. Boosting as a metaphor for algorithm design. In *Proceedings of the Ninth International Conference on Principles and Practice of Constraint Programming (CP)*, 899–903.

Leyton-Brown, K.; Nudelman, E.; Andrew, G.; McFadden, J.; and Shoham, Y. 2003b. A portfolio approach to algorithm selection. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 1542–1543.

Liberto, G. D.; Kadioglu, S.; Leo, K.; and Malitsky, Y. 2013. DASH: dynamic approach for switching heuristics. *CoRR* abs/1307.4689.

Rice, M., and Kulhari, S. 2008. A survey of static variable ordering heuristics for efficient BDD/MDD construction. *University of California Technical Report 2008.*

Shacham, O., and Yorav, K. 2006. Adaptive application of SAT solving techniques. *Electronic Notes in Theoretical Computer Science* 144(1):35–50.